Accelerating X-Ray data collection using Pyramid Beam ray casting geometries

Amir Averbuch[1] Guy Lifchitz[2] Y. Shkolnisky[3]
[1]School of Computer Science
[2] Department of Applied Mathematics, School of Mathematical Sciences
Tel Aviv University, Tel Aviv 69978
[3]Program of Applied Mathematics, Department of Mathematics
Yale University, New Haven, CT, USA

Abstract

Image reconstruction from its projections is a necessity in many applications such as medical (CT), security, inspection and others. This paper extends the 2D Fan-beam method in [2] to three dimensions. The algorithm, called Pyramid Beam (PB), is based on the parallel reconstruction algorithm in [1]. It allows fast capturing of the scanned data. In three dimensions, the reconstructions are based on the discrete X-Ray transform [1]. The PB based algorithms have different geometries where simplicity of the capturing geometries is emphasized. The PB geometries are reordered to fit parallel projection geometry. The underlying idea in the paper is to use the algorithm in [1] by porting the proposed PB geometries to fit the algorithm in [1]. The complexity of the algorithm is comparable with the 3D FFT. The results show excellent reconstruction qualities while being simple for practical use.

1 Introduction

X-Ray imaging is a critical component in many applications such as medical scans (CT, MRI, PET), baggage scanning in airports, material inspection, cars tire inspection, food inspection, biology, electronics and many more.

In practice, emitters emanating electromagnetic radiation and detectors, which measure the radiation power arrived at them, are used in X-Ray tomography. The photons radiation, which passes through the scanned object, decreases. From the radiance at the detectors and from the emitter's radiance, it is possible to reconstruct a 3D function of the radiance attenuation. The attenuation factor is unique for different materials.

With the advance of technology and the rapid increase of computational power, 3D reconstructions become practical. Using 3D reconstruction methods together with 3D visualization, greatly improve the analysis capabilities of the scanned results.

In this paper, we present several related methods to accelerate 3D X-Ray data acquisition when only one emitter is used. These methods are based on the PB geometry. Its performance is compared with the parallel beam geometry. The original (source) image is reconstructed by the application of the inverse X-Ray algorithm ([1]).

All the proposed methods in the paper are based on careful positioning of multiple detectors to enable simultaneous collection of many rays that are emitted in all directions by one emitter. Here is an hierarchical list of the methods that are described in the paper:

Centered Pyramid Beam (CPB) method has a simple geometry. The emitter and the detectors are positioned according the structure that the inverse X-Ray transform [1] dictates. It collects only portion of the actual X-Ray data. Thus, it is impractical. On the other hand, it demonstrates the capabilities of other practical methods that are described below.

Boundary Aligned emitter Pyramid Beam (BAPB) method collects all the required data to perform an image reconstruction.

Sliding Boundary Aligned emitter Pyramid Beam (SBAPB) method is a variation of the BAPB method in which the detectors are utilized more efficiently. This method reduces the number of detectors. It can be used with all the X-Ray data acquisition methods that are described here.

Mirrored Pyramid Beam (MPB) method collects only a portion of the required data for the reconstruction. The rest of the data is collected by mirroring the rays.

Distributed Pyramid Beam (DPB) is a distributed algorithm that divides the bounding volume of the object to several sub-volumes, collects the X-Ray data and performs the reconstruction of each sub-volume separately. Then, all the reconstructions are concatenated to create the complete image.

The MPB and DPB methods require that the emitter is located on planes inside the bounding volume of the object. Therefore, they are applicable to process non-solid objects, to scan simultaneously several separated objects in different X-Ray chambers, or to scan complex objects that are separated on the planes inside the bounding volume where the emitter has to be located. Any type of object (solid and non-solid) can be scanned with the BAPB method and there are no restrictions on its structure. In section 5.2, we show how to reduce the number of detectors to the minimum dictated by [1], by positioning them on moving boards. This idea is applicable to all the above methods.

In our implementation, the geometry in each axes was the same, each axes can have its own geometry.

The proposed PB ray casting topology speeds the 3D X-Ray data acquisition by $O(n^2)$ factor in comparison to the parallel beam topology.

The structure of the paper is as follows. Section 2 reviews related works on fast inversion algorithms of the X-Ray Transform while speeding data acquisition using pyramid beams or cone beams. In Section 3, the X-Ray Transform and its discrete version, which appeared in [1], are described. Section 4 demonstrates the parallel beam data acquisition and reconstruction of X-Ray data produced by an analytic parallel beam X-Ray Transform. The pyramid beam projections are defined in section 5. It contains a description of several acquisition methods and how to convert from pyramid beam projection data into parallel projection data.

2 Related works

Since memory and time complexity of the reconstruction algorithms grow polynomially with the number of dimensions then any algorithmic acceleration is critical. Acceleration can come from speeding either the acquisition or the reconstruction or from both.

Two main approaches are used to reconstruct 3D images from X-Ray projections. The first approach reconstructs separately 2D slices of the image and then concatenates the slices to form a 3D image. This requires the image to be static to prevent registration problems. It also may generate discontinuities in the reconstructed 3D image. The second approach generalizes the 2D reconstruction algorithms to any number of dimensions.

The first approach for a 3D object segmentation and reconstruction is used in [8, 9]. [10] registers the 2D slices and then reconstructs the 3D object. A technique, which improves the quality of 2D slices and then uses the improved slices to construct the 3D image via image processing methods, is described in [11, 12].

In this paper, we are interested in accelerating the acquisition while using a fast 3D X-Ray reconstruction algorithm that is described in [1]. Usually, fast 3D X-Ray reconstruction algorithms are based on the Fourier slice theorem. Some of these algorithms interpolate the polar grid into a Cartesian grid. The Fourier transform is sensitive to interpolation and the reconstructed image suffers from distortions. The filtered back projection based algorithms overcome this problem but their complexity is $O(n^4 \log n)$ where n is the image resolution in each axis. Accurate reconstruction that does not necessitate interpolation is described in [1] and it is based on the constructions in [3, 4].

Bresler et. al. [18] proposes hierarchical algorithm for applying the back projection of the 3D Radon transform. Their algorithm is a "native" 3D algorithm and does not rely on factorization of

the 3D Radon transform into pairs of 2D Radon transforms, which makes the algorithm independent of the sampling geometry. The algorithm in [18] decomposes each projection into a sum of 8 back projections each has n^2 plane-integrals projections onto $n^3/8$ volumes. Each volume is one octant of the reconstruction. The algorithms are applied recursively until each octant's size is one voxel. The complexity of the algorithm is $O(n^3 log(n))$.

Another family of reconstruction algorithms is the multi-level inversion algorithms. It divides the input sinogram to a number of subsinograms that uses either exact or approximate decomposition algorithms. The sinograms are repeatedly subdivided until they are represented by one voxel. Then, the inverse transformation is applied to reconstruct the sub-volumes. The sub-volumes are aggregated to form the final volume. An exact method to decompose the sinograms is described in [13]. This paper also presents a fast algorithm which approximates the reconstruction. In addition, it also presents a method that combines both.

Maximum likelihood expectation maximization ([14]) is an iterative reconstruction method, in which an initial reconstruction is guessed, and then updated in order to minimize the difference between the projections of the reconstructed image and the measured projections. In addition, [14] describes the cone beam data acquisition method.

An algorithm that decomposes the image frequency domain to sub-bands and reconstructs the sub-bands on a down-sampled grid is given in [17].

Cone beam projection methods, which is based on accelerating the scanned data acquisition by detecting multiple rays emitted simultaneously from a single source, are given in [14, 15, 16].

This paper proposes fast acquisition algorithm which is a variation of the cone beam method. The projection is assumed to be a collection of rays that forms a pyramid. These rays are sampled simultaneously. The reconstruction algorithm, which is described in [1], is algebraically accurate while preserving the geometric properties of the continuous transforms. It is also invertible.

3 The X-Ray Transform

The proposed fast data acquisition methods in this paper are based on the 3D X-Ray transform geometry that is described in [1]. The 3D transform is outlined here. In the rest of paper, $\vec{\mathbf{u}}$ denotes a unit vector.

The X-Ray transform of a 3D function $f = f(x, y, z)$ is a collection of all line integrals of f over all the lines in the 3D space.

Definition 3.1. *A line $l = p + t \cdot \vec{\mathbf{d}},\ t \in \mathbb{R}$ in the 3D space, is defined by its direction unit vector $\vec{\mathbf{d}} \in \mathbb{R}^3$ and a point $p \in \mathbb{R}^3$ that the line passes through.*

Definition 3.2. ***Direction by angles.*** *Two angles $\theta, \phi \in \mathbb{R}$ define a line direction by a unit vector*

$\vec{\mathbf{d}} \in \mathbb{R}^3$ *denoted* $\vec{\mathbf{d}}_{\theta,\phi}$*, by rotating the unit vector* $\vec{\mathbf{u}}_x = (1,0,0)$ *by* $-\phi$ *around the Y axis, and then rotating the resulting vector by* θ *around the Z axis.* θ *is also known as the vector heading and* ϕ *as the vector elevation. A line in* $\mathbb{R}^3$ *with direction* $\vec{\mathbf{d}}_{\theta,\phi}$ *is denoted by* $l^p_{\theta,\phi}$*.*

Definition 3.3. ***Direction by a point.*** *A point* $p_{dir} \in \mathbb{R}^3$ *defines a line direction by a unit vector, denoted by* $\vec{\mathbf{d}}_{p_{dir}}$*, as the vector from that point to the origin* $(0,0,0)$*. That is,* $\vec{\mathbf{d}} = \frac{\vec{p}_{dir}}{\|\vec{p}_{dir}\|}$*. A line in* $\mathbb{R}^3$ *with* $\vec{\mathbf{d}}_{p_{dir}}$ *is denoted by* $l^p_{p_{dir}}$*.*

Definitions 3.2 and 3.3 are equivalent.

Definition 3.4. ***Line integral.*** *The line integral of* $f(x,y,z)$ *over the line* $l^p_{\theta,\phi}$*, denoted by* $\mathcal{LI}^p_{\theta,\phi}$*, is* $\mathcal{LI}^p_{\theta,\phi} f = \int_{-\infty}^{\infty} f\left(p + t \cdot \vec{\mathbf{d}}_{\theta,\phi}\right) dt,\ \theta,\phi,t \in \mathbb{R}, p \in \mathbb{R}^3$*.*

From these definitions, the X-Ray transform of $f(x,y,z)$, denoted by $\mathcal{XR}f$, is the set

$$\mathcal{XR}f = \left\{ \mathcal{LI}^p_{\theta,\phi} f \mid \theta,\ \phi \in \mathbb{R},\ \ p \in \mathbb{R}^3 \right\}.$$

In a similar way, the line integral of $f(x,y,z)$ over the line $l^p_{p_{dir}}$, denoted by $\mathcal{LI}^p_{p_{dir}}$, is

$$\mathcal{LI}^p_{p_{dir}} f = \int_{-\infty}^{\infty} f\left(p + t \cdot \vec{\mathbf{d}}_{p_{dir}}\right) dt, \quad t \in \mathbb{R}, p, p_{dir} \in \mathbb{R}^3.$$

By using these definitions, we get that the X-Ray transform of $f(x,y,z)$, denoted by $\mathcal{XR}f$, is the set $\mathcal{XR}f = \left\{ \mathcal{LI}^p_{p_{dir}} f \mid p_{dir}, p \in \mathbb{R}^3 \right\}$.

Definition 3.5. ***Parallel projection.*** *A parallel projection of the X-Ray transform is a collection of all the computed line integrals that have the same direction. These lines are defined by a specific direction* $\vec{\mathbf{d}}_{\theta,\phi}$ *or* $\vec{\mathbf{d}}_{p_{dir}}$ *where* $\theta,\phi \in \mathbb{R}$*,* $p_{dir} \in \mathbb{R}^3$ *and an arbitrary* $p \in \mathbb{R}^3$*. The projection is denoted by* $\mathcal{XR}^p_{\theta,\phi} f$ *or* $\mathcal{XR}^p_{p_{dir}} f$*, respectively.*

The Fourier Slice Theorem links between the parallel projections $\mathcal{XR}^p_{\theta,\phi} f$, $\theta,\phi \in \mathbb{R}, p \in \mathbb{R}^3$, and the Fourier Transform. It establishes that the Fourier transform of a parallel projection in direction $\vec{\mathbf{d}}_{\theta,\phi}$ of a 3D function $f(x,y,z)$ is the Fourier transform of $f(x,y,z)$ sampled on a hyper-space perpendicular to $\vec{\mathbf{d}}_{\theta,\phi}$ that passes through the origin. Formally, $\widehat{\mathcal{XR}^p_{\theta,\phi} f} = \widehat{f}(\xi)$ where $\xi \in \left(\vec{\mathbf{d}}_{\theta,\phi}\right)^{\perp}$ is the hyper-space perpendicular to the vector $\vec{\mathbf{d}}_{\theta,\phi}$, that passes through the origin. In other words, the 2D Fourier transform of the parallel projection $\widehat{\mathcal{XR}^p_{\theta,\phi} f}$ equals to the 3D Fourier transform of $f(x,y,z)$ sampled on ξ.

Lemma 3.6. *([1]) Assume I is a discrete 3D image with resolution n in each direction. Then, the 2D FFT of a parallel projection at some direction* $\vec{\mathbf{d}}_{\theta,\phi}$*,* $\theta,\phi \in \mathbb{R}$*, is a 2D plane in the 3D FFT of the original image*

$$\begin{aligned}
\widehat{\mathcal{XR}}^p_{x,\alpha,\beta}(k,l) &= \widehat{I}(-\alpha k - \beta l, k, l) \\
\widehat{\mathcal{XR}}^p_{y,\alpha,\beta}(k,l) &= \widehat{I}(k, -\alpha k - \beta l, l) \\
\widehat{\mathcal{XR}}^p_{z,\alpha,\beta}(k,l) &= \widehat{I}(k, l, -\alpha k - \beta l)
\end{aligned}$$

where $\alpha, \beta \leq |1|$ *are the slopes between the unit vector and the cartesian axes* x, y, z, $k, l = -n, ..., n$ *and* XR_x XR_y *and* XR_z *are explained in section 3.1.*

Lemma 3.6 shows that in order to reconstruct the image from parallel projections, we have to apply the 2D Fourier transform to the projections, re-organize them in the 3D space to get the Fourier Transform of the original image, and then the 3D Inverse Fourier Transform is applied to recover the original image. The Discrete X-Ray Transform in [1] provides a $O(n^3 \log n)$ algorithm that reconstructs accurately the 3D image. It is based on the reorganization of the Fourier Transforms of the projections in the pseudo-polar grid as was explained in [3, 4].

3.1 The discrete X-Ray Transform

We assume that the data for the reconstruction is finite and discrete. The algorithm to compute the X-Ray transform of a 3D discrete image is given in [1]. The invertibility of the algorithm and its validity in representing discrete images are proven in details there. Here is a description how to discretize the image and the underlying pseudo-polar grid (see [3, 4]).

3.1.1 Discretization of the X-Ray Transform data

Following are the definitions that describe the discrete image and the sets of points defining the lines directions and their translations. We assume that the image is a discrete 3D function $f(x, y, z)$ that is defined as $f = \{f(x, y, z) | -\frac{n}{2} \leq x, y, z \leq \frac{n}{2}\}$.

According to [1], the calculated projections calculations are separated into three groups.

Definition 3.7. ***Main axis,*** *denoted* A_i*, is*

$$A_i \equiv \begin{cases} X & i = 1 \\ Y & i = 2 \\ Z & i = 3 \end{cases}.$$

Definition 3.8. ***Secondary axes,*** *denoted by* S_i^1 *and* S_i^2*, are*

$$S_i^1 \equiv \begin{cases} Y & i = 1 \\ X & otherwise \end{cases}$$

$$S_i^2 \equiv \begin{cases} Y & i = 3 \\ Z & otherwise \end{cases}.$$

The lines in $\mathbb{R}^3$ are also separated into three subsets. Each subset is associated with a main axis X, Y or Z. At each subset of lines, the absolute value of the angles between the projections of the lines on the planes XY, XZ and YZ and the main axis A_i are smaller than $45°$.

Definition 3.9. ***(Lines division I)*** *The three subsets of the lines in* $\mathbb{R}^3$ *are:*

$$\begin{aligned} L_x &= \{l^p_{\theta,\phi} | \quad -45^\circ \leq \theta < 45^\circ \text{ or } 135^\circ \leq \theta < 225^\circ, -45^\circ \leq \phi < 45^\circ \text{ or } 135^\circ \leq \phi < 225^\circ, p \in \mathbb{R}^3\} \\ L_y &= \{l^p_{\theta,\phi} | \quad 45^\circ \leq \theta < 135^\circ \text{ or } 225^\circ \leq \theta < 315^\circ, -45^\circ \leq \phi < 45^\circ \text{ or } 135^\circ \leq \phi < 225^\circ, p \in \mathbb{R}^3\} \\ L_z &= \{l^p_{\theta,\phi} | \quad 0^\circ \leq \theta < 360^\circ, 45^\circ \leq \phi < 135^\circ \text{ or } 225^\circ \leq \phi < 315^\circ, p \in \mathbb{R}^3\}. \end{aligned}$$

Loosely speaking, lines that belong to L_i are "closer" to the main axis A_i than to any other axis. This division covers all the lines in $\mathbb{R}^3$ - see proof in [1].

Lemma 3.10. *Assume that two lines pass through the same point* $p \in \mathbb{R}^3$*. The first line direction is defined by* $\phi, \theta \in \mathbb{R}$*. The second line direction is defined by* $\phi + 180^\circ$ *and* θ *or by* $-\phi$ *and* $\theta + 180^\circ$*. Then,* $l^p_{\theta,\phi} \equiv l^p_{\theta,\phi+180^\circ} \equiv l^p_{\theta+180^\circ,-\phi}$.

From Lemma 3.10, definition 3.9 becomes:

Definition 3.11. ***(Lines division II)*** *The three subsets of the lines in* $\mathbb{R}^3$ *are:*

$$\begin{aligned} L_x &= \{l^p_{\theta,\phi} | \quad -45^\circ \leq \theta < 45^\circ, -45^\circ \leq \phi < 45^\circ, p \in \mathbb{R}^3\} \\ L_y &= \{l^p_{\theta,\phi} | \quad 45^\circ \leq \theta < 135^\circ, -45^\circ \leq \phi < 45^\circ, p \in \mathbb{R}^3\} \\ L_z &= \{l^p_{\theta,\phi} | \quad 0^\circ \leq \theta < 360^\circ, 45^\circ \leq \phi < 135^\circ, p \in \mathbb{R}^3\}. \end{aligned}$$

For a spatially bounded function $f(x, y, z)$, the majority of the lines in $\mathbb{R}^3$ do not intersect the function's bounding volume. As explained in definition 3.4, line integral is defined by a direction and a point the line passes through. The limitations on the directions of the lines, which participate in the discrete X-Ray transform, are defined in 3.11. Lemma 3.12 determines the minimal set of points required to define lines that produce non-trivial line integrals.

Lemma 3.12. ***Points of interest ([1])*** *Assume that each of the coordinates* x, y, z *of the function* $f(x, y, z)$ *are spatially bounded by the interval* $[-n/2, n/2]$*. In addition, we restrict the directions to the set defined in 3.11. Then, the minimal set of points, which is required to define the non-trivial line integrals, includes points with coordinate* $A_i = 0$ *and the coordinates* S^1_i *and* S^2_i *are bounded by* $[-n, n]$*. The lines pass through these points as was described in definition 3.4.*

Definition 3.13. ***Discrete set of points of interest.*** *A discrete subset of points, which were defined in Lemma 3.12, have the coordinates* $A_i = 0$ *and* $S^1_i, S^2_i \in \{-n, ..., n\}$*. This set is denoted by* P^i_{tr}.

The collection of non-trivial line integrals over lines from L_i, $i = 1, 2, 3$, is denoted by XR_x XR_y and XR_z, respectively.

For a point $p_{dir} \in \mathbb{R}^3$, definition 3.3 describes how to determine the line direction. In order to discretize the lines sets in definition 3.11, a discrete set of points is defined.

Definition 3.14. ***Current lines*** L_i *are defined for each* A_i *by:*

$$L_i = \begin{cases} L_x & i = 1 \\ L_y & i = 2 \\ L_z & i = 3 \end{cases} .$$

Definition 3.15. ***Discrete set of directions.*** *A discrete set of points, denoted by* P^i_{dir}*, which includes points with the coordinate* $A_i = n/2$ *and* $S^1_i, S^2_i \in \{-n/2, ..., n/2\}$*, defines a discrete subset of the line set* L_i*,* $i = 1, 2, 3$*.*

For all the points in P^i_{tr}, the coordinate A_i is equal to 0. Similarly, for all the points in P^i_{dir}, A_i is equal to -1. Therefore, these points can be defined uniquely by pairs of values from the other two coordinates S^1_i and S^2_i.

Definition 3.16. ***Simplified directions and translations sets.*** *The points in the sets* P^i_{tr} *and* P^i_{dir} *can be defined uniquely by the pairs* (s, t) *and* (k, l)*, respectively,* $s, t, k, l \in \mathbb{R}$*.* k *and* s *represent the coordinate* S^1_i *while* l *or* t *represent the coordinate* S^2_i*. These sets of pairs are denoted by* $\tilde{P}^i_{tr}$ *and* $\tilde{P}^i_{dir}$*, respectively.*

According to [1], the discrete sets P^i_{tr} and P^i_{dir}, $i = 1, 2, 3$, define the exact set of line integrals required to arrange the data on the pseudo-polar grid. This enables to use the fast and accurate reconstruction method that was described there.

4 Reconstruction from analytical parallel projections

The validity and the accuracy of the reconstruction algorithms were tested and verified on the 3D Shepp-Logan image which was constructed analytically. In this section, the reconstruction algorithm uses parallel projections. In section 5, the Pyramid-beam reconstruction is described.

Two methods are used to compute the projections:

Method 1: The parallel projections arrays are computed analytically.

Method 2: The parallel projections arrays are computed using the discrete X-Ray transform ([1]).

Table 4.1 displays the l_2 error between the two methods that compute the projections. The computations that are marked by * were done by a distributed algorithm - see section 6.

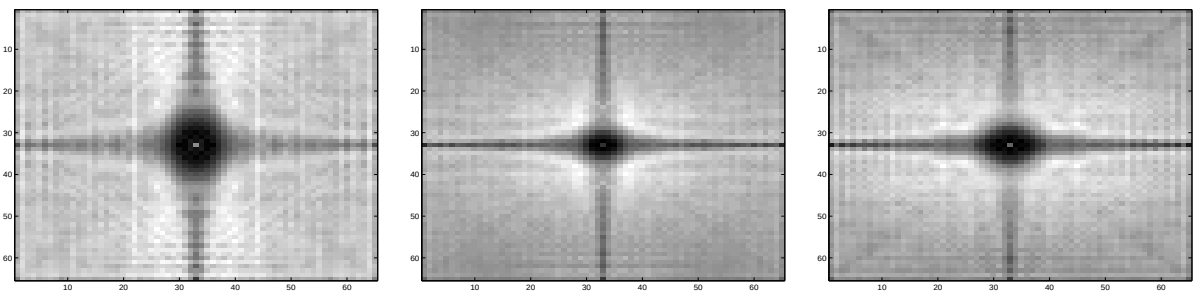
n	l_2 error	l_2/n^3 error per voxel
4	5.0291	8.28e-4
8	62.770	8.938e-4
16	266.434	2.823e-4
32	1186.664	8.6e-5
64	4294.527	2.036e-5
128*	4896.590	1.485e-6
256*	11418.536	2.19e-7

Table 4.1: The l_2 error between the analytic and the discrete computation ([1]) of the projections of the Shepp-Logan image.

Table 4.1 shows that the computational error per voxel decreases as the resolution of the image increases.

Figure 4.1 displays the l_2 error between projections of the Shepp-Logan image at different directions where $n = 64$. The minimum and the maximum of the l_2 errors were 21.463 and 47.88, respectively. Each pixel in the image is the l_2 of the difference between the projections computed at different directions according to definition 3.3. The error per pixel is derived from the difference between the analytic computation and the exact X-Ray computation from [1]. The minimum and the maximum error values show that the contribution of each line integral to the l_2 error is small as shown in table 4.1.

From Fig. 4.1, we see that when at least one of the secondary coordinates S_i^1 and S_i^2 of the direction points p_{dir}^i is close to zero, the error decreases. This is due to the fact that the points coordinates along the line integral are proportional to the tan of the angle. Therefore, the error from a line integral computation increases with the angle.



(a) (b) (c)

Figure 4.1: l_2 error between projections when the main axes is X (a), Y (b) and Z (c)

4.1 Parallel projection geometry

The discrete parallel projections in respect to a main axis A_i are retrieved by restricting the line integrals from definition 3.5 to the set of lines defined by the points in $\tilde{P}^i_{tr}$ and $\tilde{P}^i_{dir}$ (see definition 3.16). For a point $\tilde{p}^{\,i}_{dir} \in \tilde{P}^i_{dir}$, the discrete parallel projection $\mathsf{XR}^{\tilde{P}^i_{tr}}_{i,\tilde{p}^{\,i}_{dir}} f$, $i = 1, 2, 3$, contains line integrals whose directions are defined by $\tilde{p}^{\,i}_{dir}$. For each point $\tilde{p}^{\,i}_{tr} \in \tilde{P}^i_{tr}$, there is exactly one line integral in the projection that passes through the point $\tilde{p}^{\,i}_{tr}$.

The image is bounded in the interval $[-1, 1]$ in each axis. The image resolution at each axis is n. This implies that the set of coordinates $\{-n/2, ..., n/2\}$ is mapped to $[-1, 1]$. The points in the set P^i_{tr}, defined in 3.13, have the coordinates $A_i = 0$ and $S^1_i, S^2_i \in \{2j/n | j = -n, ..., n\}$. The points in the set P^i_{dir}, defined in 3.15, have the coordinates $A_i = 1$ and $S^1_i, S^2_i \in \{2j/n | j = -n/2, ..., n/2\}$.

In order to understand where the emitter and detector have to be placed, a specific line is analyzed.

Definition 4.1. ***Generalized point description.*** *$p^i_G(u, v, w)$ is a point where $u, v, w \in \mathbb{R}$ are the coordinates of A_i, S^1_i and S^2_i, respectively.*

Definition 4.2. ***Generalized planes.*** *A plane, which is defined by setting the main axis coordinate to a constant value $A_i \equiv c$, $c \in \mathbb{R}$, is denoted by $P_i(c)$.*

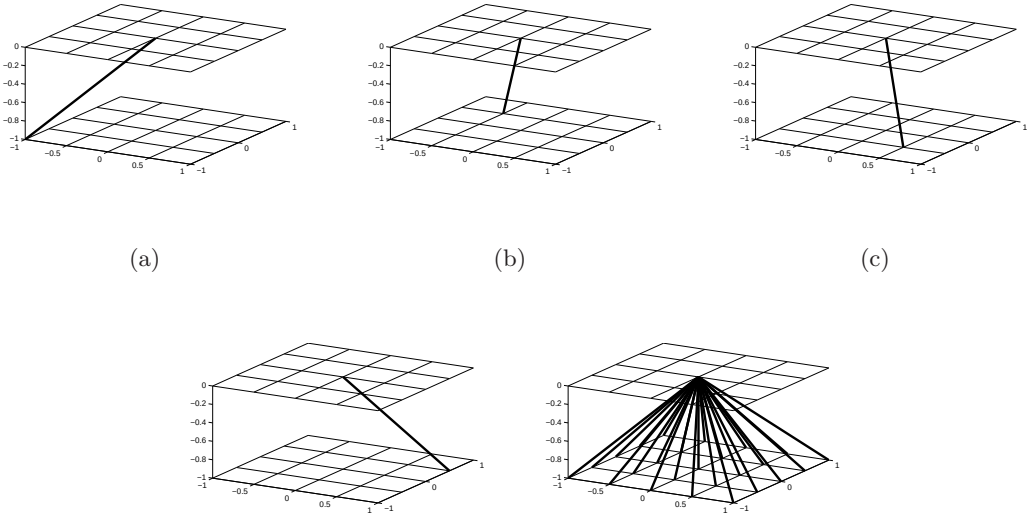
A line that is defined by the translation point $\tilde{p}^{\,i}_{tr}(s, t)$ and by the direction point $\tilde{p}^{\,i}_{dir}(k, l)$ passes through the point $p^i_G(0, s, t)$. From definition 3.3, the line direction is $\frac{\vec{p}^{\,i}_G(-1,k,l)}{\|\vec{p}^{\,i}_G(-1,k,l)\|}$. Therefore, this line intersects the planes $P_i(1)$ and $P_i(-1)$ at the points $p^i_G(1, s + k, t + l)$ and $p^i_G(-1, s - k, t - l)$, respectively, where $s, t \in \{2j/n | j = -n, ..., n\}$ and $, k, l \in \{2j/n | j = -n/2, ..., n/2\}$.

All the line integrals in the discrete parallel projection $\mathsf{XR}^{\tilde{P}^i_{tr}}_{i,\tilde{p}^{\,i}_{dir}} f$, where $\tilde{p}^{\,i}_{dir}$ is a specific point in $\tilde{P}^i_{dir}$ and $\tilde{p}^{\,i}_{tr}$, are all the points in $\tilde{P}^i_{tr}$ that have the same direction. Each line passes through a different point on the plane $P_i(0)$. Therefore, the lines are parallel as this method name suggests.

For a specific direction defined by the point $\tilde{p}^{\,i}_{dir}(k, l) \in \tilde{P}^{\,i}_{dir}$, the process, which calculates the projection $\mathsf{XR}^{\tilde{P}^i_{tr}}_{i,\tilde{p}^{\,i}_{dir}(k,l)} f$ using one emitter, is described below. For each point $\tilde{p}^{\,i}_{tr}(s, t) \in \tilde{P}^i_{tr}$, the emitter is placed at the point $p^i_G(1, s + k, t + l)$ and the detector is placed at the point $p^i_G(-1, s - k, t - l)$, $k, l \in \{2j/n | j = -n/2, ..., n/2\}$ and $s, t \in \{2j/n | j = -n, ..., n\}$. The emitter's positions are all in a square where coordinates $A_i = 1$ and S^1_i, S^2_i are from the interval $[-3, 3]$. The detectors' positions are the same as the emitter except $A_i = -1$. This geometry shows that the emitter and the detector are being located on parallel planes $P_i(1)$ and $P_i(-1)$, respectively.

Line integrals in the same parallel projection cannot be calculated simultaneously. Therefore, only one detector is needed to compute this projection.

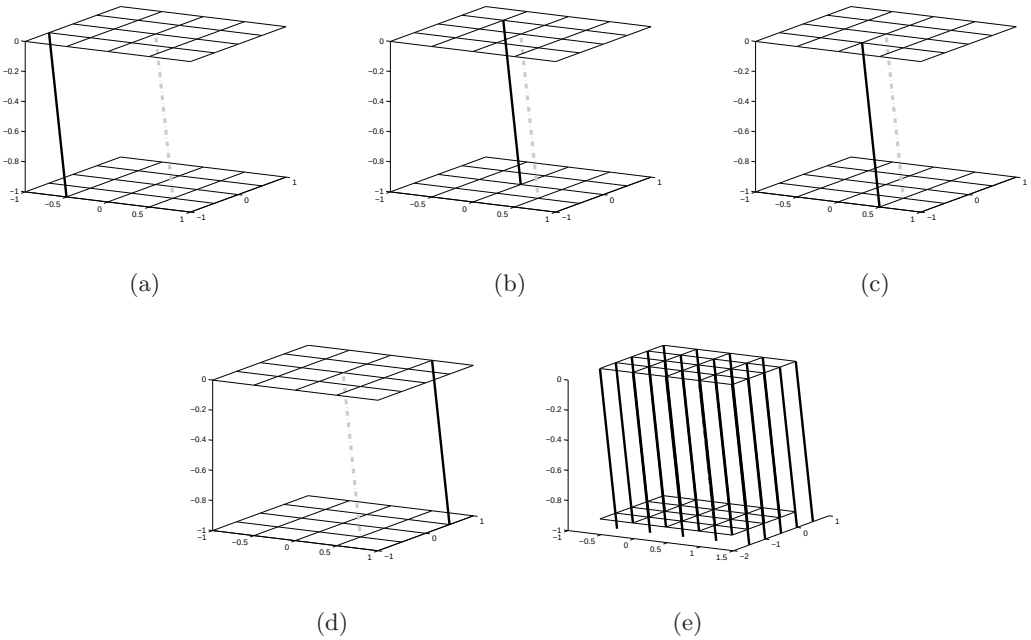
Figure 4.2 shows the lines from L_3 (see definition 3.14), which are defined by $\tilde{p}^3_{tr}(0, 0)$ and by different points from $\tilde{P}^3_{dir}$. Figure 4.3 shows the lines from L_3 which are defined by $\tilde{p}^3_{dir}(0.5, -0.5)$ and by different points from $\tilde{P}^3_{tr}$. In Fig. 4.3, the gray dashed line is defined by the translation $\tilde{p}^3_{tr}(0, 0)$.



(a) (b) (c)

(d) (e)

Figure 4.2: Lines defined by the point $\tilde{p}^3_{tr}(0,0)$ and by different $\tilde{p}^3_{dir} \in \tilde{P}^3_{dir}$. (a) Line defined by $\tilde{p}^3_{dir}(-1,-1)$. (b) Line defined by $\tilde{p}^3_{dir}(-0.5,0.5)$. (c) Line defined by $\tilde{p}^3_{dir}(0.5,-0.5)$. (d) Line defined by $\tilde{p}^i_{dir}(1,0.5)$. (e) All lines directions defined by the set $\tilde{P}^3_{dir}$

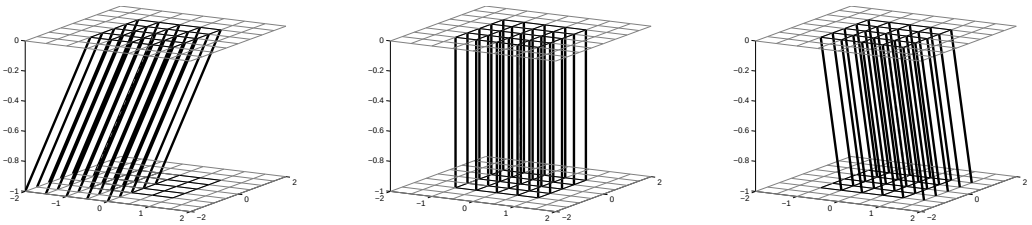


(a) (b) (c)

(d) (e)

Figure 4.3: Lines defined by $\tilde{p}^3_{dir}(0.5,-0.5)$ and by different $\tilde{p}^3_{tr} \in \tilde{P}^3_{tr}$. (a) Line defined by $\tilde{p}^3_{tr}(-1.0,-0.5)$. (b) Line defined by $\tilde{p}^3_{tr}(-0.5,0.5)$. (c) Line defined by $\tilde{p}^3_{tr}(0.0,-0.5)$. (d) Line defined by $\tilde{p}^3_{tr}(0.5,1.0)$. (e) A subset of lines from the parallel projection $XR^{\tilde{P}^3_{tr}}_{3,\tilde{p}^3_{dir}(0.5,-0.5)}f$

Figure 4.4 describes several subsets of lines from parallel projections at different directions.



(a) (b) (c)

Figure 4.4: Parallel projections. (a) A subset of the projection $XR^{\tilde{P}^3_{tr}}_{3,\tilde{p}^{\,3}_{dir}(-1.0,-1.0)}f$. (b) A subset of the projection $XR^{\tilde{P}^3_{tr}}_{3,\tilde{p}^{\,3}_{dir}(0.0,0.0)}f$. (c) A subset of the projection $XR^{\tilde{P}^3_{tr}}_{3,\tilde{p}^{\,3}_{dir}(0.5,0.0)}f$

According to Lemma 3.12 and the fact that $f = f(x, y, z)$ is bounded in each direction, it is easy to verify that line integrals over lines with translation greater than 2.0 in one of the dimensions are equal to 0.

Figure 4.5 describes the parallel projection $XR^{\tilde{P}^3_{tr}}_{3,\tilde{p}^{\,3}_{dir}(1.0,1.0)}f$. The bold lines in Fig. 4.5 represent the bounding volume of $f(x, y, z)$. The gray lines begin in plane $P_i(1)$ where the emitter is located and ends in the plane $P_i(-1)$ where the detector is located. The figure shows that the emitter's and the detector's coordinates S^1_i and S^2_i, respectively, are in the interval $[-3, 3]$, as was mentioned in this section before.

From definition 3.13, $\tilde{p}^{\,3}_{tr} \in \tilde{P}^3_{tr}$ are points on the plane $P_3(0)$. Figure 4.5 shows that the coordinates S^1_i and S^2_i of the points on the plane $P_3(0)$ are in the interval $[-2.0, 2.0]$. It also shows that lines, which are defined by the points $\tilde{p}^{\,3}_{tr}(s, t)$ where s or t equal -2.0 or 2.0, are tangent to the bounding volume. If $|s| > 2$ or $|t| > 2$ then the lines will not intersect the bounding volume. These results are also true for projections in directions defined by $\tilde{p}^{\,3}_{dir}(k, l)$ where $|k| < 1$ or $|l| < 1$.

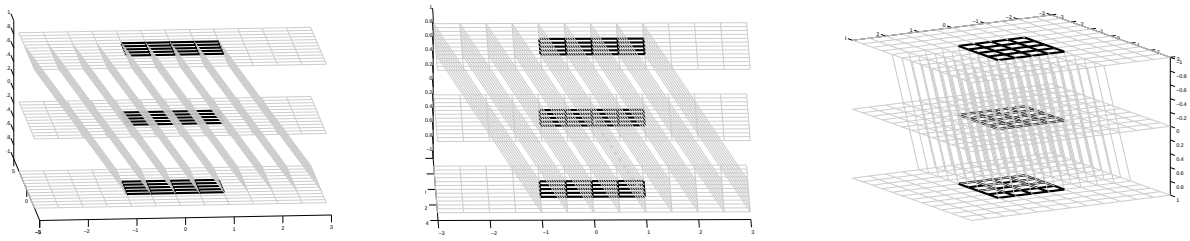


Figure 4.5: Different views of non-trivial lines that have the same direction $\tilde{p}^{\,3}_{dir}(1.0, 1.0)$. The emitter is located on the plane $P_3(1)$ and the detector is located on the plane $P_3(-1)$.

The inverse discrete X-Ray transform ([1]) reconstructs the image from a set of parallel projections. The sets P^i_{tr} and P^i_{dir} together define all the line integrals required to reconstruct the image.

Definition 4.3. ***The input to the inverse X-Ray transform.*** *The input for the inverse X-Ray transform is all the parallel projections defined by the sets P^i_{tr} and P^i_{dir}. This set, denoted PP, is:*

$$PP = \left\{ XR^{p^i_{tr}}_{i,p^i_{dir}} f | p^i_{dir} \in P^i_{dir}, \quad p^i_{dir} \in P^i_{dir}, \quad i = 1,2,3 \right\}.$$

The parallel projection $XR^{\tilde{p}^i_{tr}}_{i,\tilde{p}^i_{dir}} f$ is computed for each direction defined by the point $\tilde{p}^i_{dir} \in \tilde{P}^i_{dir}$. The projection is a 2D array of size $(2n+1) \times (2n+1)$. The coordinates of each element in the array correspond to a point $\tilde{p}^i_{tr} \in \tilde{P}^i_{tr}$. The value of the array element is $LI^{\tilde{p}^i_{tr}}_{i,\tilde{p}^i_{dir}} f$.

Definition 4.4. ***Parallel projections data structure.*** *All the line integrals, required to reconstruct the image by the discrete inverse X-Ray transform ([1]), are stored in the array PP_{ds}. The first coordinate in the array is i, $i = 1,2,3$. It represents the main axis X, Y or Z. The following two coordinates $p, q \in \{1, ..., n+1\}$ represent the direction of the line integral $\tilde{p}^i_{dir}(k,l)$ where $k = (p - n/2 - 1) \cdot 2/n$ and $l = (q - n/2 - 1) \cdot 2/n$. The last two coordinates, $u, v \in \{1, ..., 2n+1\}$, represent the translation of the line integral, $\tilde{p}^i_{tr}(s,t)$ where $s = (u - n - 1) \cdot 2/n$ and $t = (v - n - 1) \cdot 2/n$. Formally,*

$$PP_{ds}(i,p,q,u,v) = LI^{\tilde{p}^i_{tr}(s,t)}_{i,\tilde{p}^i_{dir}(k,l)} f. \tag{4.1}$$

For specific i, p, q and all $u, v \in \{1, ..., 2n+1\}$, the collection of $PP_{ds}(i,p,q,u,v)$ values is the parallel projections $XR^{\tilde{P}^i_{tr}}_{i,\tilde{p}^i_{dir}(k,l)} f$ where $k = (p - n/2 - 1) \cdot 2/n$ and $l = (q - n/2 - 1) \cdot 2/n$.

In order to compute a parallel projection in a given direction, the emitter and the detector have to be positioned at $(2n+1)^2$ locations. It means that each parallel projection requires $(2n+1)^2$ operations. For each main axis A_i, $i = 1,2,3$, there are $(n+1)^2$ parallel projections that correspond to different directions. Thus, computing the data structure PP_{ds} requires $3\,(2n+1)^2\,(n+1)^2$ operations. Therefore, the total number of operations is $O(n^4)$ where n is the resolution of each dimension.

The images were computed by two methods: 1. Analytically. 2. The projections were computed analytically and the reconstruction was done via the application of the inverse discrete X-Ray transform ([1]). The differences between the projections computed by 1. and 2. were computed. The numerical results from this reconstruction are presented in section 5.4.

5 Pyramid-Beam (PB) reconstruction

Emitters are more expensive than detectors. Therefore, the PB data acquisition geometry suggests to add detectors in order to collect simultaneously the line integrals in multiple directions.

Usually, the use of only one emitter is common in X-Ray tomography to reduce the data acquisition costs. PB geometry is used in X-Ray transform when one emitter is present. It becomes more efficient for data acquisition than parallel geometry. In PB geometry based data acquisition, the line integrals

in all directions can be calculated simultaneously. Therefore, the number of operations required to collect the projected data has to be divided by $O(n^2)$.

In this section, a family of methods, which are based on PB geometry, is described. For two constants $c_1, c_2 \in \mathbb{R}, c_1 \neq c_2$, PB projections are computed by locating the emitter on the plane $P_i(c_1)$ (see definition 4.2) and the detectors on the plane $P_i(c_2)$. For different PB methods, c_1 and c_2 have different values. This geometry allows a simultaneous computation of multiple different line integrals that pass through the same point and have different directions.

The planes $P_i(c_1)$ and $P_i(c_2)$ are orthogonal. Therefore, the lines participating in each PB projection form a shape of a square pyramid (see Fig. 5.1). PB projection is defined in a similar way to definition 3.5.

Definition 5.1. ***PB projection.*** *A PB projection of the X-Ray transform is a collection of all the computed line integrals that pass through a specific point $p \in \mathbb{R}^3$ and have arbitrary directions $\vec{\mathbf{d}}_{\theta,\phi}$ or $\vec{\mathbf{d}}_{p_{dir}}$ where $\theta, \phi \in \mathbb{R}, p_{dir} \in \mathbb{R}^3$. This projection is denoted by $\mathsf{XR}^p_{\theta,\phi} f$ or by $\mathsf{XR}^p_{p_{dir}} f$.*

A PB projection $\mathsf{XR}^{\tilde{p}^i_{tr}}_{i,\tilde{P}^i_{dir}} f$, $i = 1, 2, 3$, is a collection of line integrals defined by a specific point from the set $\tilde{p}^i_{tr}$ and by all the points from the set $\tilde{p}^{\,i}_{dir} \in \tilde{P}^i_{dir}$ (see definition 3.16).

The main goal of this paper is to find an efficient method to collect simultaneously multiple line integrals. In order to reconstruct the image by the inverse X-Ray transform ([1]), the PB projections have to be transformed into the PP_{ds} data structure defined by Eq. 4.1. This transformation is called reordering. Each data acquisition method has its own version of reordering algorithm.

The idea is that the algorithm in [1] is efficient and accurate and each acquisition method with different PB geometries has to be transformed into a parallel projection methodology described in definition 3.5. The reordering algorithm algorithm does not necessitate any operations.

Several PB methods called CPB, $BAPB$, $SBAPB$ and MPB are presented here (see also section 1). For each method, its data acquisition geometry and its reordering algorithm are described and its complexity is analyzed.

5.1 Centered Pyramid-beam (CPB) acquisition geometry

The CPB geometry is based on the sets $\tilde{P}^i_{tr}$ and $\tilde{P}^i_{dir}$ (see definition 3.16). Assume $s, t \in \{2j/n | j = -n, ..., n\}$ and $i = 1, 2, 3$. For each point $\tilde{p}^i_{tr}(s,t) \in \tilde{P}^i_{tr}$, the emitter is located at the point $p^i_G(0, s, t)$ (see definition 4.1). The detectors are located at the points $p^i_G(-1, s-k, t-l)$ where $k, l \in \{2j/n | j = -n/2, ..., n/2\}$.

The emitter's positions are located at points from the set $\tilde{P}^i_{tr}$. According to Lemma 3.12 and the fact that $f(x, y, z)$ is bounded by $[-1, 1]$ at every axis, these positions are all located in a square with coordinate $A_i = 0$, and the coordinates of S^1_i and S^2_i are taken from the interval $[-2, 2]$. Therefore,

the detectors' positions have the coordinate $A_i = -1$ and the coordinates S_i^1 and S_i^2 in the interval $[-3, 3]$. It means that on the plane $P_i(-1)$ there are $(3n+1)^2$ detectors. For each emitter's position $p_G^i(0, s, t)$, only $(n+1)^2$ detectors, whose secondary coordinates S_i^1 and S_i^2 vary between $[s-1, s+1]$ and $[t-1, t+1]$, respectively, are of interest. These detectors' values are line integrals of lines from the set PP (see definition 4.3). The other detectors' values are not needed by the inverse discrete X-Ray transform. Their values are line integrals of lines with directions that do not belong to P_{dir}^i.

According to definition 5.1, in order to compute a CPB projection $XR_{i,\tilde{P}_{dir}^i}^{\tilde{p}_{tr}^i(s,t)} f$ for the point $\tilde{p}_{tr}^i(s,t) \in \tilde{P}_{tr}^i$, $s, t \in \{2j/n | j = -n, ..., n\}$, the emitter is placed at the point $p_G^i(0, s, t)$. The projection is an array that contains the values of the detectors with coordinate $A_i = -1$. S_i^1 coordinates are in the interval $[s-1, s+1]$ and S_i^2 coordinates are in the interval $[t-1, t+1]$.

For each point $\tilde{p}_{tr}^i \in \tilde{P}_{tr}^i$, a CPB projection is computed. The projection's result is a 2D array of size $(n+1) \times (n+1)$. The coordinates of each element in the array correspond to a point $\tilde{p}_{dir}^i \in \tilde{P}_{dir}^i$ and its value is $LI_{i,\tilde{p}_{dir}^i}^{\tilde{p}_{tr}^i} f$.

Definition 5.2. *CPB **data structure.** All the line integrals, which are required to reconstruct the image by the discrete inverse X-Ray transform ([1]), are stored in the array CPB_{ds}. The first coordinate is i, $i = 1, 2, 3$. It represents the main axis X, Y or Z. The following two coordinates, $u, v \in \{1, ..., 2n+1\}$, represent the translation of the line integral $\tilde{p}_{tr}^i(s,t)$ where $s = (u - n - 1) \cdot 2/n$ and $t = (v - n - 1) \cdot 2/n$. The last two coordinates, $p, q \in \{1, ..., n+1\}$, represent the direction of the line integral $\tilde{p}_{dir}^i(k,l)$ where $k = (p - n/2 - 1) \cdot 2/n$ and $l = (q - n/2 - 1) \cdot 2/n$. Formally,*

$$CPB_{ds}(i, u, v, p, q) = LI_{i,\tilde{p}_{dir}^i(k,l)}^{\tilde{p}_{tr}^i(s,t)} f. \tag{5.1}$$

For specific i, u, v and all $p, q \in \{1, ..., n+1\}$, the collection of the values $CPB_{ds}(i, u, v, p, q)$ is the CPB projection $XR_{i,\tilde{p}_{dir}^i}^{\tilde{p}_{tr}^i(s,t)} f$ where $s = (u - n - 1) \cdot 2/n$ and $t = (v - n - 1) \cdot 2/n$.

Definitions 4.4 and 5.2 describe the coordinates of the data structures of PP_{ds} and CPB_{ds}, respectively. Lemma 5.3 shows how to switch between these two data structures. The reordering algorithm switches between pairs of coordinates in the CPB_{ds} data structure.

Lemma 5.3. ***Reordering the** CPB **data structure.** The data in the CPB_{ds} data structure is reordered into parallel projections by $PP_{ds}(i, p, q, u, v) = CPB_{ds}(i, u, v, p, q)$, where $i = 1, 2, 3$, $p, q \in \{1, ..., n+1\}$ and $u, v \in \{1, ..., 2n+1\}$. The pairs (p, q) represent the line integrals directions $\tilde{p}_{dir}^i(k,l)$, $k = (p - n/2 - 1) \cdot 2/n$ and $l = (q - n/2 - 1) \cdot 2/n$. The pairs (u, v) represent the line integrals translations $\tilde{p}_{tr}^i(s,t)$, $s = (u - n - 1) \cdot 2/n$ and $t = (v - n - 1) \cdot 2/n$.*

Proof. The range of the forth and fifth coordinates in PP_{ds} is the same as the range of the second and third coordinates in CPB_{ds}. Each element in the data structure $PP_{ds}(i, p, q, u, v)$ contains the line integral $LI_{i,\tilde{p}_{dir}^i(k,l)}^{\tilde{p}_{tr}^i(s,t)} f$ (see definition 3.4), where $k = (p - n/2 - 1) \cdot 2/n$, $l = (q - n/2 - 1) \cdot 2/n$, $s = (u - n - 1) \cdot$

$2/n$ and $t = (v-n-1)\cdot 2/n$. From definition 5.2, each element in the data structure $CPB_{ds}(i,u,v,p,q)$ contains the same line integral. Therefore, the equality $PP_{ds}(i,p,q,u,v) = CPB_{ds}(i,u,v,p,q)$ holds for every selection of u, v. □

In order to compute a CPB projection, the emitter has to be located at one point. For each main axis A_i, $i = 1, 2, 3$, there are $(2n+1)^2$ CPB projections that correspond to different line translations. The reordering does not require any operations as was mentioned at the beginning of this section. Therefore, the computation of the data structure CPB_{ds} requires $3\,(2n+1)^2$ operations, which sums to $O(n^2)$ operations. Thus, it accelerates the parallel data acquisition geometry by $O(n^2)$.

Figure 5.1 visualizes the line integrals that participate in the CPB projections from different emitter's positions.

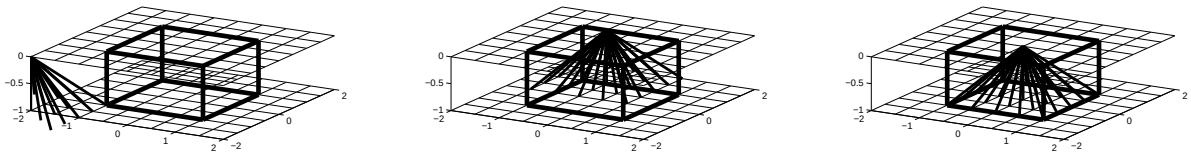


Figure 5.1: Locations of he emitter in the CPB geometry. (a) Emitter located at $\tilde{p}^3_{tr}(-2.0, -2.0)$. (b)Emitter located at $\tilde{p}^3_{tr}(-0.5, 1.0)$. (c) Emitter located at $\tilde{p}^3_{tr}(0.0, 0.0)$

Usually, the scanned objects are solid. They are placed around the origin. This prevents from placing the emitter on the planes $P_i(0)$ in contradiction to the CPB guidelines. Even if the object is neither solid nor placed in the origin, we still cannot utilize the CPB methodology to reconstruct the image. The reason for that is the fact that the outputs from line integrals represent only portion from the lines between the planes $P_i(0)$ and $P_i(-1)$. Thus, they do not represent the whole line integral through the scanned object.

In order to overcome these problems, section 5.2 proposes a method where the emitter is placed on the planes $P_i(1)$. Section 5.3 on the other hand, proposes a method that calculates line integrals on both sides of the emitter's location by adding detectors on the plane $P_i(1)$. This overcomes only the second problem.

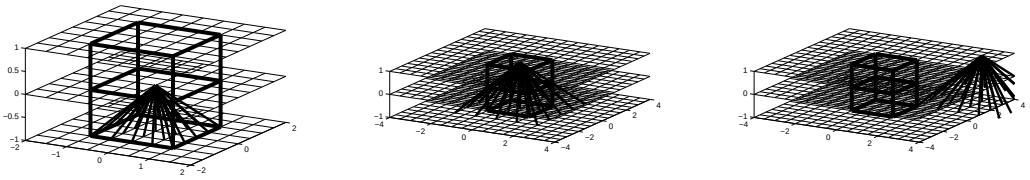
5.2 Boundary aligned Pyramid-beam (BAPB) acquisition geometry

As mentioned in section 5.1, the CPB method assumes that the emitter is placed on the planes $P_i(0)$ (see definition 4.2). This limits the scenarios for which this method is useful. In order to overcome this limitation, the $BAPB$ proposes to move the emitter's location to the boundary of the object being

scanned. Specifically, it recommends to place the emitter on the planes $P_i(1)$. This displacement affects the geometry and the projections' results. This also dictates a change to be made in the reordering algorithm.

To detect line integrals with different directions that pass trough the same point in the $BAPB$ geometry, the emitter is located at different positions in the plane $P_i(1)$. Multiple detectors are located on equally spaced grid in a square in the plane $P_i(-1)$. Then, only a subset of the detectors' values, which correspond to line integrals whose tangents are bounded by $[-1, 1]$, are stored in the $BAPB$ data structure (see definition 5.8).

Figure 5.2 displays the transformation that is applied to the CPB emitter's locations and its effect on the pyramid geometry where (a) is the original geometry of the CPB and (b) and (c) are the geometries of PAPB. It is possible to see that the tip of the pyramid in (a) is on the plane $P_i(0)$ and in (b) and (c) P_1. Moreover, the pyramid base in (b) and (c) is twice the size in (a).



(a) CPB projection (b) $BAPB$ projection (c) Emitter located at $p^3_G(1.0, 3.0, 3.0)$

Figure 5.2: Different emitter's positions in the $BAPB$ geometry.

The two points $\tilde{p}^{\,i}_{tr}(s,t) \in \tilde{P}^i_{tr}$ and $\tilde{p}^{\,i}_{dir}(k,l) \in \tilde{P}^i_{dir}$ define a line. This line intersects the plane $P_i(1)$ at the point $p^i_G(1, s+k, t+l)$. A second line with a different direction $\tilde{p}^{\,i}_{dir}(q,r)$, which passes through the point $p^i_G(1, s+k, t+l)$, intersects the planes $P_i(0)$ and $P_i(-1)$ at the points $p^i_G(0, s+k-q, t+l-r)$ and $p^i_G(-1, s+k-2\cdot q, t+l-2\cdot r)$, respectively. Therefore, the second line is defined by the points $\tilde{p}^{\,i}_{tr}(s+k-q, t+l-r)$ and $\tilde{p}^{\,i}_{dir}(q,r)$. Since $-1 \leq q, r \leq 1$, the detectors' secondary coordinates satisfy $s+k-2 \leq S^1_i \leq s+k+2$ and $t+l-2 \leq S^2_i \leq t+l+2$, $-2 \leq s,t \leq 2$ and $-1 \leq k,l \leq 1$. Therefore, the locations of the detectors, which are required to collect the line integrals in the set PP, have the coordinates $A_i = -1$, $-5 \leq S^1_i, S^2_i \leq 5$. The emitter's positions have the coordinates $A_i = 1$, $-3 \leq S^1_i, S^2_i \leq 3$.

This geometry requires to position the emitter in $(3n+1)^2$ locations while spreading $(5n+1)^2$ detectors on the plane $P_i(-1)$. From these $(5n+1)^2$ detectors, only $(n+1)^2$ detectors values represent line integrals from the set PP. How to select these detectors? Two points from the set $\tilde{P}^i_{dir}$ define two different line directions. Two line integrals with different line directions, which pass through the same point, will be detected by different detectors. The distance between these two detectors in the

$BAPB$ geometry is twice the distance between the corresponding detectors in the CPB geometry.

Lemma 5.4. ***Distance between detectors.*** *Assume we have two rays with different directions $\tilde{p}^i_{dir}(k_1, l_1)$ and $\tilde{p}^i_{dir}(k_2, l_2) \in \tilde{P}^i_{dir}$ that are emitted from the same emitter. The distance between the detectors, which detect these rays in the $BAPB$ geometry, is twice the distance in the CPB geometry.*

Proof. The proof is based on triangles similarity. In order to simplify triangles height calculation, assume that the S^1_i coordinates of the two detectors and the emitter are the same. According to the CPB geometry, the emitter is located at a point on the plane $P_i(0)$. When the $BAPB$ method is applied, the emitter is located on the plane $P_i(1)$. The lines directions are the same in both geometries. Therefore, the locations of the emitter and the two detectors, in both methods, construct two similar triangles. From these assumptions, it is easy to calculate the triangles heights. The CPB triangle height equals 1 while the $BAPB$ triangle height is 2. Since these triangles are similar, the proportion between the heights is also the proportion between the triangles edges. For both triangles, the edges on the plane $P_i(-1)$ are the distances between the detectors. Therefore, the distance in the $BAPB$ geometry is doubled. In a similar way, these triangles similarity can be shown for every two detectors and emitter which compute the line integrals with the same two directions in both geometries. □

Lemma 5.5. ***Positioning the emitter and the step size.*** *In order to collect the line integrals given in PP (see definition 4.3), the distance between two neighboring locations of the emitter should be the same as in the CPB geometry.*

Proof. For two lines in a parallel projection, which are defined by the points $\tilde{p}^i_{tr}(s, t)$, $\tilde{p}^i_{tr}(s + 1/n, t)$ and $\tilde{p}^i_{dir}(k, l)$, the emitter in the CPB geometry must be located at $p^i_G(0, s, t)$ and $p^i_G(0, s + 1/n, t)$. In the $BAPB$ geometry, the emitter must be located at $p^i_G(1, s+k, t+l)$ and $p^i_G(1, s+k+1/n, t+1)$. Therefore, the distance between the two points in the CPB geometry and the two points in the $BAPB$ geometry is the same. □

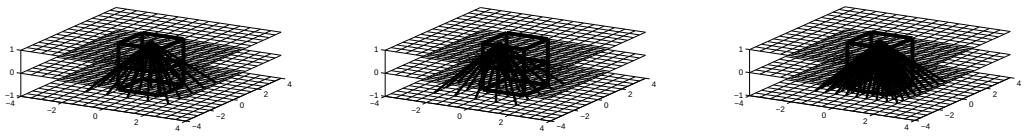
Figure 5.3 visualizes Lemma 5.5. The emitter is located at two neighboring locations. The distance between the closest detectors, which contain line integrals from PP in the two $BAPB$ pyramid projections, is the same as in the CPB and as in the parallel geometries.

By comparing between the $BAPB$ and the parallel projections geometries, we get that the emitter in both methods is located on the planes $P_i(1)$ with S^1_i and S^2_i coordinates satisfying $-3 \leq S^1_i, S^2_i \leq 3$.

The $BAPB$ geometry together with Lemmas 5.4 and 5.5, lead to the following conclusion:

Corollary 5.6. ***Inefficient detectors utilization.*** *The detectors are poorly utilized in the $BAPB$ method. At each location of the emitter, only $(n + 1)^2$ of the $(5n + 1)^2$ detectors, are line integral values from the set PP. Moreover, there are no two neighboring detectors which contain values from PP. Either odd or even positioned detectors are used for the reconstruction.*

Figure 5.3 visualizes Corollary 5.6. In Figs . 5.3(a) Fig. 5.3(b) the detectors are placed in the even and odd positions, respectively. Figure 5.3(c) shows all the detectors that collect line integrals whose tangents are bounded by $[-1, 1]$.



(a) (b) (c)

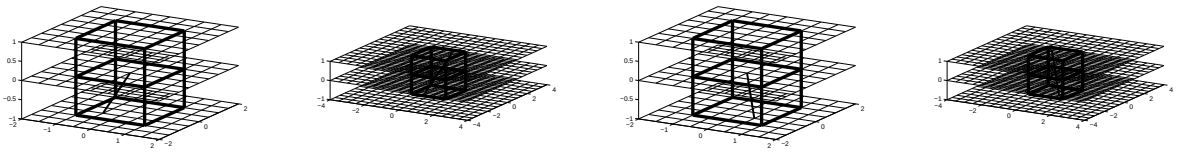
Figure 5.3: Detectors' locations for two adjacent emitter's locations in the $BAPB$ geometry. (a) Emitter located at $p^3_G(1.0, 0.0, 0.0)$. (b) Emitter located at $p^3_G(1.0, -0.5, -0.5)$. (c) All line integrals whose tangents are bounded by $[-1, 1]$

As was mentioned before, the line integral defined by the points $\tilde{p}^i_{tr}(s,t) \in \tilde{P}^i_{tr}$ and $\tilde{p}^i_{dir}(k,l) \in \tilde{P}^i_{dir}$, intersects the plane $P_i(1)$ at the point $p^i_G(1, s+k, t+l)$. This leads to the following conclusion:

Corollary 5.7. ***Lines from*** PP***, which pass through the same points, appear in different*** $BAPB$ ***pyramid-beam projections.*** *Two line integrals, which are defined by two different directions* $\tilde{p}^i_{dir}(k_1, l_1)$ *and* $\tilde{p}^i_{dir}(k_2, l_2)$ *and by one translation* $\tilde{p}^i_{dir}(s,t)$*, appear in different* $BAPB$ *projections. These line integrals will appear in the projections where the emitter is located at the points* $p^i_G(1, s + k_1, t + l_1)$ *and* $p^i_G(1, s + k_2, t + l_2)$.

This differs from the CPB method, in which both line integrals appear in the same projection where the emitter is located at $p^i_G(0, s, t)$.

Figure 5.4 visualizes Corollary 5.7. It shows two lines with a translation that is defined by $\tilde{p}^3_{tr}(0, 0)$. The lines directions are defined by $\tilde{p}^3_{dir}(-0.5, -0.5)$ and $\tilde{p}^3_{tr}(0.5, -0.5)$. In the CPB geometry (left), both line integral are acquired by the same pyramid (centered at $(0, 0)$). In the $BAPB$ geometry (right), each line is acquired by a different $BAPB$ pyramid.



(a) (b) (c) (d)

Figure 5.4: The translation of the emitter's location in $BAPB$ geometry. (a) A line defined by $\tilde{p}^3_{tr}(0.0, 0.0)$ and $\tilde{p}^3_{dir}(-0.5, -0.5)$ in the CPB geometry. (b) A line defined by $\tilde{p}^3_{tr}(0.0, 0.0)$ and $\tilde{p}^3_{dir}(-0.5, -0.5)$ in the $BAPB$ geometry. (c) A line defined by $\tilde{p}^3_{tr}(0.0, 0.0)$ and $\tilde{p}^3_{dir}(0.5, -0.5)$ in the CPB geometry. (d) A line defined by $\tilde{p}^3_{tr}(0.0, 0.0)$ and $\tilde{p}^3_{dir}(0.5, -0.5)$ in the $BAPB$ geometry

The translation of the emitter in the $BAPB$ geometry does not enable to compute the projections in a similar way as was defined in 5.1. Instead, the emitter is located in positions where $A_i = 1$, and S^1_i, S^2_i vary in the interval $[-3, 3]$ with the step $1/n$. For each $S^1_i = s$ and $S^2_i = t$, the detectors, whose coordinates are $s - 2 \leq S^1_i \leq s + 2$ and $t - 2 \leq S^2_i \leq t + 2$, generate the projection.

The projection's result is a 2D array of size $(2n + 1) \times (2n + 1)$. The coordinates of each element in the array correspond to a pair (k, l) where $k, l \in \{j/n | j = -n, ..., n\}$. This pair represents the direction of the line integral in the same way as the points in the set $\tilde{P}^i_{dir}$. This pair together with the emitter's position $p^i_G(1, s, t)$, $s, t \in \{2j/n | j = -3n/2, ..., 3n/2\}$, define the pair $(s - k, t - l)$. The pair $(s - k, t - l)$, represents the emitter's translation point $p^i_G(0, s - k, t - l)$. The points $\tilde{p}^i_{dir}(k, l)$ and $\tilde{p}^i_{tr}(s-k, t-l)$, which are defined by k, l, s and t, can neither be in $\tilde{P}^i_{dir}$ nor in $\tilde{P}^i_{tr}$ (see definition 3.16), respectively. This is due to the fact that half of the detectors' values do not represent line integrals from the set PP (see definition 4.3). The value of the array element is $\mathsf{LI}^{\tilde{p}^i_{tr}(s-k,t-l)}_{i,\tilde{p}^i_{dir}(k,l)} f$ (see definition 3.4).

Definition 5.8. *$BAPB$ **data structure.** All the line integrals, computed by the $BAPB$ projections, are stored in $BAPB_{ds}$. Its first coordinate is i, $i = 1, 2, 3$. It represents the main axis X, Y or Z. The following two coordinates, $u, v \in \{1, ..., 3n + 1\}$ represent the translation of the line integral $\tilde{p}^i_{tr}(s, t)$ where $s = (u-3n/2-1)\cdot 2/n$ and $t = (v-3n/2-1)\cdot 2/n$. The last two coordinates $p, q \in \{1, ..., 2n+1\}$ represent the direction of the line integral $\tilde{p}^i_{dir}(k, l)$ where $k = (p - n - 1)/n$ and $l = (q - n - 1)/n$. Formally,*

$$BAPB_{ds}(i, u, v, p, q) = \mathsf{LI}^{\tilde{p}^i_{tr}(s,t)}_{i,\tilde{p}^i_{dir}(k,l)} f. \tag{5.2}$$

For specific i, u, v and all $p, q \in \{1, ..., 2n + 1\}$, the collection of values $BAPB_{ds}(i, u, v, p, q)$ is the $BAPB$ projection $\mathsf{XR}^{\tilde{p}^i_{tr}(s,t)}_{i,\tilde{p}^i_{dir}} f$ where $s = (u - 3n/2 - 1) \cdot 2/n$ and $t = (v - 3n/2 - 1) \cdot 2/n$.

From Corollary 5.7 and the $(2n+1)^2$ size of each projction in $BAPB$, we get that a new reordering algorithm for processing efficiently the $BAPB$ projections is needed.

Lemma 5.9. ***Reordering of the BAPB dataset.*** *The data in the $BAPB_{ds}$ is reordered to fit the parallel projections geometry by $PP_{ds}(i,p,q,u,v) = BAPB_{ds}(i, u-p+n+1, v-q+n+1, 2\cdot p-1, 2\cdot q-1)$, where $i = 1,2,3$, $p,q \in \{1,...,n+1\}$ and $u,v \in \{1,...,2n+1\}$. The pair (p,q) represents the parallel projections directions $\tilde{p}^i_{dir}(k,l)$ where $k = (p-n/2-1)\cdot 2/n$ and $l = (q-n/2-1)\cdot 2/n$. The pair (u,v) represents the parallel projections translation $\tilde{p}^i_{tr}(s,t)$, where $s = (u-n-1)\cdot 2/n$ and $t = (v-n-1)\cdot 2/n$.*

Proof. A line with the direction $\tilde{p}^i_{dir}(k,l)$, which passes through the point $\tilde{p}^i_{tr}(s,t)$, intersects the plane $P_i(1)$ at the point $p^i_G(1, s-k, t-l)$. Therefore, the coordinates $u,v \in \{1,...,2n+1\}$ are transformed to the coordinates in $BAPB_{ds}$ representing the position $p^i_G(1, s-k, t-l)$. The range of the translation indices in PP_{ds} is $[1, 2n+1]$. The range of the emitter's coordinates S^1_i and S^2_i is $[-2, 2]$. The range of the directions indices in PP_{ds} is $[1, n+1]$. The tangents range are bounded by $[-1, 1]$. A linear mapping of these ranges shows that $s = 2(u-1-n)/n$ and $k = 2(p-1-n/2)/n$. Therefore, $s-k = 2(u-1-n)/n - 2(p-1-n/2)/n = 2/n(u-p-n/2)$. Translation of the range $[-3,3]$ of the emitter's positions back to the range $[1, 3n+1]$ of the indices, shows that the new index, which represents the coordinate S^1_i of the emitter, is $2/n(u-p-n/2)\cdot n/2 + 3n/2 + 1 = u-p+n+1$. The second coordinate is transformed similarly. The indices, which represent the $BAPB_{ds}$ direction, are doubled and then 1 is subtracted since only the odd indices of the projection, belong to the set of line integrals in PP participate as was shown in Corollary 5.6. □

For each main axis A_i there are $(3n+1)^2$ $BAPB$ projections. At each emitter's location, all $(5n+1)^2$ line integrals are measured simultaneously. Only $(n+1)^2$ line integrals from each projection are used. The reordering is assumed to take no time. Therefore, computing the data structure $BAPB_{ds}$ requires $3\,(3n+1)^2$ operations, i.e. $O(n^2)$ operations. Even if the computations of CPB_{ds} and $BAPB_{ds}$ take $O(n^2)$ operations, the $BAPB_{ds}$ computation requires about 9/4 times more operations while using about 25/9 times more detectors.

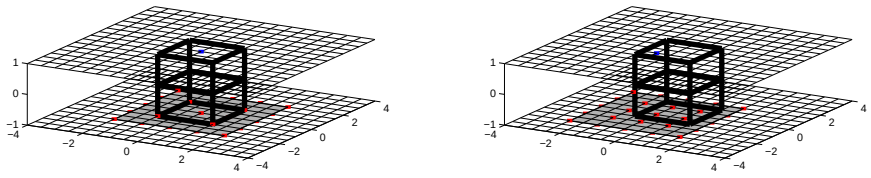
In order to overcome the low detectors utilization in the $BAPB$ method, a variation of the method is suggested. This variation uses only $(n+1)^2$ detectors. These detectors are located on a moving board. The distances between the detectors are doubled in order to collect only line integrals from the set PP. In order to collect the correct data when the emitter moves to its next position, the board with the detectors moves together with the emitter. Therefore, the detectors' coordinates S^1_i and S^2_i change at the same amount as the emitter's coordinates S^1_i and S^2_i.

This setting reduces the number of required detectors by a factor of 25 and thus it provides a full utilization of the detectors.

The CPB geometry also utilizes the detectors inefficiently. At each projection, only $(n+1)^2$ from $(3n+1)^2$ detectors on the planes $P_i(-1)$ are used. A similar variation can be applied to the CPB method to reduce the number of required detectors by a factor of 9. In this setting, the distance

between detectors stays the same as the original distance.

This setting is called Sliding Boundary Aligned Pyramid Beam ($SBAPB$). Figure 5.5 shows detectors (marked in red) which are placed on a moving board (marked as gray rectangle). The detectors' distances are doubled. The right figure shows how the board moves together with the emitter (marked in blue).



(a) (b)

Figure 5.5: Sliding detectors positioned on a mobile board that moves together with the emitter. (a) Emitter located at $p_G^3(1.0, 0.0, 0.0)$ in the $SBAPB$ geometry. (b) Emitter located at $p_G^3(1.0, -0.5, -0.5)$ in the $SBAPB$ geometry

When $SBAPB$ is used, the projections become $(n+1)\times(n+1)$ arrays. All the data elements in these arrays contain valuable data. The reordering transform becomes: $PP_{ds}(i, p, q, u, v) = BAPB_{ds}(i, u - p + n + 1, v - q + n + 1, p, q)$ where i, p, q, u and v are the same as in Lemma 5.9.

The time complexity of the $SBAPB$ data acquisition method is the same as the $BAPB$ complexity since there is no difference between these methods except for the number of line integral being calculated simultaneously. The memory complexity is also $O(n^2)$ but it is reduced by a factor 25.

5.3 Mirrored Pyramid Beam (MPB) acquisition geometry from multiple objects

The CPB geometry does not allow to move the emitter on the planes $P_i(0)$ when the scanned object is solid. This problem does not exist for multiple objects that are placed neither in chambers around the main axes nor for scanned non-solid objects. On the other hand, as mentioned in section 5.1, only portion of the line integrals between the planes $P_i(0)$ and $P_i(-1)$ is computed (see Fig. 5.6(a)).

Here we extend the CPB geometry that overcomes this problem. Another set of detectors is placed on the planes $P_i(1)$. This set represents the mirror image of the original set in respect to planes $P_i(0)$. The rays emitted from the emitter, which are detected by this new set of detectors, form a mirror image of the original pyramid (the gray pyramid in Fig. 5.7(a)). Due to the symmetry of the original pyramid, each line integral in the original pyramid has its line extension in the mirrored pyramid. The

sum of the line integrals is the complete line integral through the scanned object (see Figs. 5.6(b) and 5.6(c)).

Two CPB projections are computed for each point $p^i_{tr} \in P^i_{tr}$. One projection uses the original set of detectors and the other uses the new set of detectors. Each projection's result is a 2D array of size $(n+1)\times(n+1)$. The coordinates of each element in the arrays correspond to a point $\tilde{p}^{\,i}_{dir}(k,l) \in \tilde{P}^i_{dir}$. The value of an array element in the projection, computed by the original CPB pyramid (black pyramid in Fig. 7(a)), is $L\!I^{\tilde{p}^{\,i}_{tr}}_{i,\tilde{p}^{\,i}_{dir}(k,l)} f^-_i$, where f^-_i is a portion of the function $f(x,y,z)$ between the planes $P_i(0)$ and $P_i(-1)$. Similarly, the value of the array element in the projection, which was computed by the mirrored pyramid (gray pyramid in fig. 7(a)), is $L\!I^{\tilde{p}^{\,i}_{tr}}_{i,\tilde{p}^{\,i}_{dir}(k,l)} f^+_i$, where f^+_i is a portion of the function $f(x,y,z)$ between the planes $P_i(0)$ and $P_i(1)$.

Definition 5.10. MPB ***data structure.*** *All the line integrals, which are required to reconstruct the image by the discrete inverse X-Ray transform [1], are stored in the arrays MPB^-_{ds} and MPB^+_{ds}. The first coordinate in each array, $i = 1,2,3$, represents the main axis X, Y or Z. The following two coordinates $u, v \in \{1,...,2n+1\}$ represent the translation of the line integral, $\tilde{p}^{\,i}_{tr}(s,t)$ where $s = (u-n-1)\cdot 2/n$ and $t = (v-n-1)\cdot 2/n$. The last two coordinates, $p, q \in \{1,...,n+1\}$ represent the direction of the line integral $\tilde{p}^{\,i}_{dir}(k,l)$ where $k = (p-n/2-1)\cdot 2/n$ and $l = (q-n/2-1)\cdot 2/n$. Formally,*

$$\begin{aligned} MPB^-_{ds}(i,u,v,p,q) &= L\!I^{\tilde{p}^{\,i}_{tr}(s,t)}_{i,\tilde{p}^{\,i}_{dir}(k,l)} f^-_i. \\ MPB^+_{ds}(i,u,v,p,q) &= L\!I^{\tilde{p}^{\,i}_{tr}(s,t)}_{i,\tilde{p}^{\,i}_{dir}(k,l)} f^+_i. \end{aligned} \tag{5.3}$$

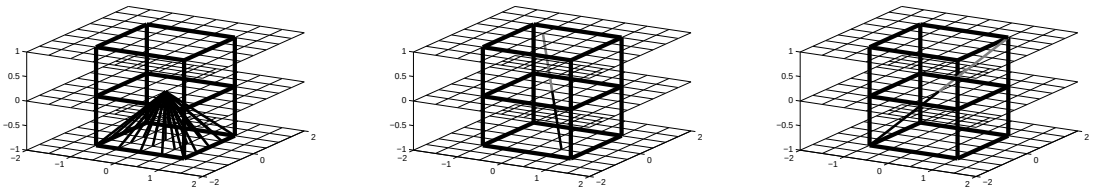
For specific i,u,v and all $p,q \in \{1,...,n+1\}$, the collection of $MPB^-_{ds}(i,u,v,p,q)$ values is the CPB projection $XR^{\tilde{p}^{\,i}_{tr}(s,t)}_{i,\tilde{p}^{\,i}_{dir}} f$.

Lemma 5.11. ***Reordering the*** MPB ***data structure.*** *The data in the MPB^-_{ds} and MPB^+_{ds} data structures is reordered into parallel projections by $PP_{ds}(i,p,q,u,v) = MPB^-_{ds}(i,u,v,p,q) + MPB^+_{ds}(i,u,v,p,q)$ where $i = 1,2,3$, $p,q \in \{1,...,n+1\}$ and $u,v \in \{1,...,2n+1\}$. The pair (p,q) represents the parallel projections directions $\tilde{p}^{\,i}_{dir}(k,l)$. $k = (p-n/2-1)\cdot 2/n$ and $l = (q-n/2-1)\cdot 2/n$. The pair (u,v) represents the line integrals translations $\tilde{p}^{\,i}_{tr}(s,t)$. $s = (p-n-1)\cdot 2/n$ and $t = (q-n-1)\cdot 2/n$.*

Collecting the line integrals with the MPB method requires $O(n^2)$ operations. $O(n^2)$ additions are required to compute the full line integrals through f. In contrast to other methods described in this paper, the reordering algorithm does take time since after the data acquisition, two portions of each line are added. The memory complexity stays $O(n^2)$ while the number of detectors is doubled.

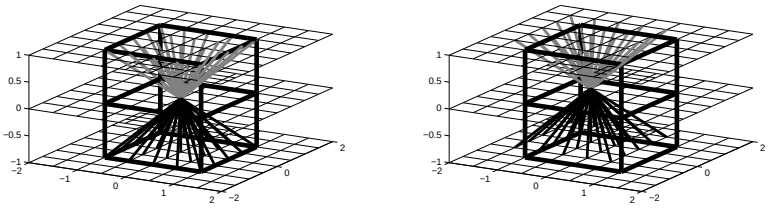
The MPB geometry is based on the sets of points $\tilde{P}^i_{tr}$ and $\tilde{P}^i_{dir}$ (see definition 3.16). This method can be used to scan simultaneously eight objects with lower resolutions. Putting an object in one of

the eight chambers and doubling the number of detectors placed in each plane, can be a substitute for the $BAPB$ method in the sense that this method can be applied to any type of scanned object - solid or non-solid. When it is known that chambers are kept empty, it is possible to reduce significantly the number of the required detectors.



(a) (b) (c)

Figure 5.6: Line integrals calculated with the MPB geometry. Black lines are calculated by the original CPB pyramid, and gray lines are calculated by MPB. (a) CPB scanning of only half of the object. (b) Line integrals defined by $\tilde{p}^3_{dir}(-0.5, 0.5)$ and $\tilde{p}^3_{tr}(0.0, 0.0)$. (c) Line integrals defined by $\tilde{p}^3_{dir}(-1.0, -1.0)$ and $\tilde{p}^3_{tr}(0.0, 0.0)$



(a) (b)

Figure 5.7: MPB geometry. (a) Emitter is located at $\widetilde{p}^3_{tr}(0.0, 0.0)$. (b) Emitter is located at $\widetilde{p}^3_{tr}(-0.5, 0.5)$

5.4 Numerical results

The performance of the reconstruction algorithm that uses different acquisitions strategies (described in sections 5.1-5.3) is showed in this section. The 3D Shepp-Logan with different resolutions were reconstructed by two different methods: 1. Analytic calculation - straightforward approach. 2. The projections were calculated analytically and then the 3D inverse X-Ray transform ([1]) was applied to reconstruct the 3D object. The l_2 error between these two methods is computed.

Five different geometries were described in the paper. PP, CPB and $BAPB$ are the only ones that have substantial different geometries. The CPB geometry was implemented by computing all the

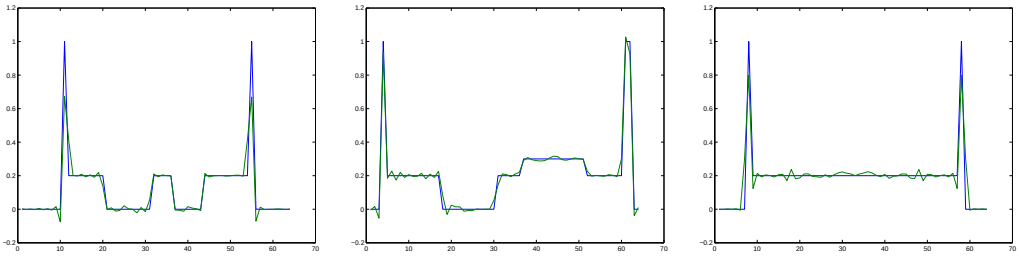
line integrals between the planes $P_i(1)$ and $P_i(-1)$ that pass through the object. These computations also fit MPB line integrals calculations. The outputs were not separated into two different arrays as was done in the original MPB method.

The numerical outputs from the application of PP, CPB and $BAPB$ methods were almost identical. They demonstrate the convergence of the reconstructed image to its analytic image version as the image's resolution increases. The time and memory requirements increases by power of 3 with resolution increase. Therefore, the algorithms were tested on images with resolutions $n = 4, 8, 16, 32, 64$. For higher resolutions such as $n = 128, 256$, a distributed algorithm was implemented (see section 6).

Table 5.1 shows the decrease of the reconstruction error as the image resolution n in each direction increases. Different acquisition methods generate almost identical results. Therefore, Table 5.1 presents only the results from the $BAPB$ method. Figure 5.8 shows the profiles of the main axes of the reconstructed image in comparison to the analytic reconstruction.

n	$BAPB\ l_{2(diff)}$	$BAPB\ \frac{l_{2(diff)}}{l_{2(analytic)}}$
4	0.9	0.491
8	5.182	0.709
16	13.270	0.626
32	36.777	0.485
64	33.372	0.318
128*	68.838	0.231
256*	145.902	0.173

Table 5.1: $l_{2(diff)}$ is the l_2 norm of the difference between the image that was reconstructed from the analytic projections using [1] and the analytic image. The projections were calculated with the $BAPB$ geometry. $l_{2(analytic)}$ is the l_2 norm of the analytic image.



(a) Profile of a line from the center of the image where y and z are constant

(b) Y middle profile

(c) Z middle profile

Figure 5.8: Comparison between line profiles of the reconstructed image (uses projections that were computed analytically and reconstructed by the application of [1]) and the analytic image. The line profiles are $(x, n/2, n/2)$, $(n/2, y, n/2)$ and $(n/2, n/2, x)$, where $x, y, z = 1, \ldots, n$. The solid lines and the dotted lines represent the line profiles from the analytic image and from the reconstructed image, respectively, for $n = 64$.

5.5 Conclusions

In sections 5.1 - 5.3, we described a family of reconstruction algorithms that acquire the scanned data via different PB geometries. These geometries accelerate the data acquisition for the X-Ray reconstruction that uses the method in [1]. All these methods save $O(n^2)$ operations in the data acquisition process by measuring simultaneously line integrals in different directions. All the described geometries are independent of the axes. In our implementation, the geometry in each axes was the same. In other implementation, each axes can have its own geometry. For example, if an object does not intersect the plane where one of the axes is zero, then, we can apply the MPB method only along this axis while preserving the resolution of the other two axes by applying the $BAPB$ method to them. Moving boards with detectors can be used in each PB method. This can further reduce the data acquisition costs. Only the PP, CPB and $BAPB$ methods where implemented and tested in this paper. It is reasonable to expect that the other methods ($SBAPB$ and MPB) will have the same performance by producing almost identical numerical accuracies. This is due to the fact the data is the same while the ordering is different.

6 Distributed algorithm

The time and memory complexity of the algorithm increase by power 3 with the image resolution. To speedup the computation, we present a distributed algorithm. This algorithm divides the bounding box, which contains the object, to d^3 volumes where each box contains parts of the scanned object. The distribution factor d is the number of sections each axis is divided into. Each volume is bounded

by six planes from the set of planes $P_i(c)$ where $c \in \{2j/d - 1 | j = 1, ..., d-1\}$. The resolution of each volume in each dimension is n/d. The data acquisition and reconstruction of each volume is done independently using any of the methods described in section 5. Once all the volumes are reconstructed, they are concatenated into one object. The analytic line integral calculation had to be modified in order to compute a line integral only between the clipping planes $P_i(c)$ that bound the reconstructed volume. Instead of finding the distance between the intersection points of the line and the ellipsoids, the line integral calculation finds the minimal distance between the intersection points of the line with the ellipsoids and with the planes that bound the reconstructed volume.

The number of measured ray directions depends on the volumes resolutions. Since this resolution degraded to n/d in each computational box, the distributed algorithm calculates line integrals over a limited set of directions compared to the non-distributed algorithm. Explicitly, there are $(n/d)^2$ different directions compared with n^2 different directions in a non-distributed algorithm. Another disadvantage of the algorithm is its independency between different sub-volumes reconstruction, which creates discontinuities between the sub-volumes on the connecting planes $P_i(c)$, $c \in \{2j/d - 1 | j = 1, ..., d-1\}$. In order to reduce this discontinuity, a smoothing algorithm is applied to lines between the concatenated images. The smoothing algorithm works in stages. Each stage smooths the lines along one axis. The result of each stage is the input image for the next smoothing stage. We start with the X axis. Define $P \triangleq \{jn/d | j = 1, ..., d-1\}$. The smoothed image along the $X-$axis is:

$$SI_x(p,q,r) = \begin{cases} -\frac{1}{16}I(p-3,q,r) + \frac{9}{16}I(p-1,q,r) + \frac{9}{16}I(p+1,q,r) - \frac{1}{16}I(p+3,q,r) & p \in P \\ I(p,q,r) & \text{else} \end{cases}.$$

In a similar way, the smoothed image along the Y axis is:

$$SI_y(q,p,r) = \begin{cases} -\frac{1}{16}SI_x(q,p-3,r) + \frac{9}{16}SI_x(q,p-1,r) + \frac{9}{16}SI_x(q,p+1,r) - \frac{1}{16}SI_x(q,p+3,r) & p \in P \\ SI_x(q,p,r) & \text{else} \end{cases}$$

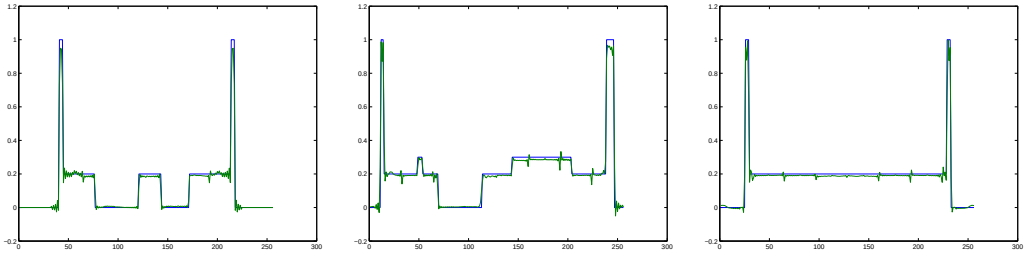
and the smoothed lines along the Z axis is:

$$SI_z(q,r,p) = \begin{cases} -\frac{1}{16}SI_y(q,p-3,r) + \frac{9}{16}SI_y(q,r,p-1) + \frac{9}{16}SI_y(q,r,p+1) - \frac{1}{16}SI_y(q,r,p+3) & p \in P \\ SI_y(q,r,p) & \text{else} \end{cases}$$

where SI_x, SI_y and SI_z are the smoothed images along the axes X, Y and Z, respectively. I is the reconstructed image and $p, q, r \in \{1, ..., n\}$.

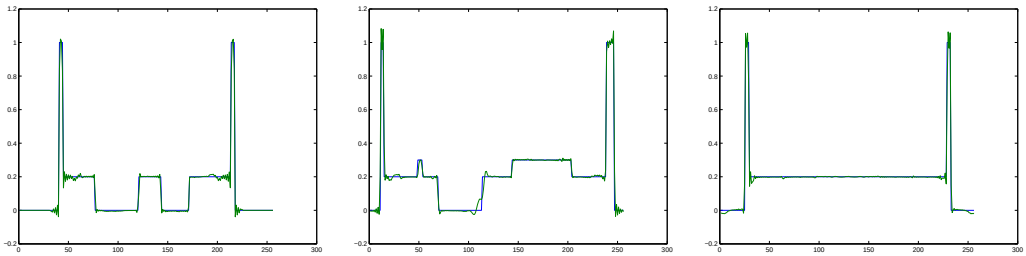
The smoothing algorithm on the connecting lines of the image can be used more than once to increase the smoothing. The discontinuities in the connecting lines on the planes $P_i(c)$ are seen in the profiles of the reconstructed image that is shown in Fig. 6.1. Figure 6.2 shows how the smoothing algorithm eliminates the discontinuities. The smoothing operation recalculates the values of the image on the planes $P_i(c)$ where $c \in \{2j/d - 1 | j = 1, ..., d-1\}$. When applying the smoothing operation

to a specific axis, the values of the planes $P_i(c)$, which are perpendicular to that axis, are being recalculated. The conclusion is that the application of the smoothing operation along a specific axis should be applied only when the derivative of the image in the direction of this axis is small in a region around the planes which are recalculated.



(a) Middle profile in X (b) Middle profile in Y (c) Middle profile in Z

Figure 6.1: Comparison between line profiles of the reconstructed image ([1]) and the analytic image. The line profiles are $(x, 0, 0)$, $(0, y, 0)$ and $(0, 0, z)$ where $x, y, z = 1, \ldots, n$. The solid profile lines were taken from from the analytic image and the dotted lines were taken from the lines profiles of the reconstructed image. The connecting lines in the reconstructed image were not smoothed. $n = 256$. We can see the discontinuities where the distributed implementation separated the volumes into pieces.



(a) Smoothed middle profile from X (b) Smoothed middle profile from Y (c) Smoothed middle profile from Z

Figure 6.2: Comparison between the line profiles that were taken from the reconstructed image ([1]) and the analytic image. The line profiles are $(x, 0, 0)$, $(0, y, 0)$ and $(0, 0, z)$ where $x, y, z = 1, ...n$. Line profiles taken from the analytic image are denoted by solid lines. Line profiles taken from the reconstructed image are denoted by dotted lines. The boundary lines in the reconstructed image were smoothed. $n = 256$. The smoothing operation eliminated the discontinuities from the boundary lines.

The distributed algorithm reduces time and memory requirements in the reconstruction. On the other hand, it degrades the ray directions resolution and causes discontinuity on the planes $P_i(p)$.

7 Conclusions

The algorithms in the paper can be improved in several ways. Simultaneous data acquisition in multiple subspaces can be improved since the methodologies in this paper describe a repeated separable application of the algorithms to each of the main axes A_i. But some of the data in different axes A_i can be acquired simultaneously. This can save time while reducing the area the emitter has to move on. This can also reduce the physical dimensions of the inspected machine. It is also possible to show that some line integrals equal other line integrals in another subset. Sharing these values can improve the algorithm's efficiency.

Two approaches to achieve distributed calculations were described. The first in section 6 requires to change the analytic calculations of the projections. The second approach divides the volume into sub-volumes and then to use the original set of rays and to calculate the contribution of each of the sub-volumes to each of the original line integral. This will impose several changes in the numeric calculation of the transform and in the calculation of the inverse transform ([1]). The analytic calculations of line integrals have to take into consideration the clipped planes of the sub-volumes. Both methods suffer from boundaries errors between sub-volumes since the reconstruction lacks interpolating data along these boundaries. On the other hand, distributed computation can improve the performance of the algorithms. The first approach is impractical for solid objects since it requires placing the emitters and the detectors along the boundaries of the sub-volumes. Therefore, a new way to compute the inverse of the distributed 3D discrete X-Ray transform with the original set of line integrals should be developed.

References

[1] A. Averbuch, Y. Shkolnisky, 3D Discrete X-Ray Transform, Applied Computational Harmonic Analysis, vol. 17, pp. 259–276, 2004.

[2] A. Averbuch, I. Sedelnikov, Y. Shkolnisky, CT reconstruction from parallel and fan-beam projections by 2D discrete Radon transform, submitted.

[3] A. Averbuch, R. Coifman, D. Donoho, Y. Shkolnisky, A Framework for Discrete Integral Transformations I the Pseudo-polar Fourier transform, SIAM Journal of Scientific Computing, 30(2), pp. 764–784, 2008.

[4] A. Averbuch, R. Coifman, D. Donoho, Y. Shkolnisky, I. Sedelnikov, A Framework for Discrete Integral Transformations II 2D discrete Radon transform, SIAM Journal of Scientific Computing, 30(2), pp. 785–803, 2008.

[5] S. R. Deans, The Radon Transform and Some of Its Applications, New York: Wiley, 1983.

[6] F. Natterer, The Mathematics of Computerized Tomography. New York: Wiley, 1986.

[7] A. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, New York: IEEE Press, 1988.

[8] A. Schenk, G. Prause, and H. O. Peitgen, Efficient Semiautomatic Segmentation of 3D Objects, Medical Images, Vol. 1935/2000, pp. 71–131, 2000.

[9] K. S. Shreedhara and S. P. Indira, Construction of 3-D Objects Using 2-D Cross Sectional Data and Curves, Advanced Computing and Communications, pp. 630–631, 2006.

[10] S. Krinidis, C. Nikou, and I. Pitas, 3D Volume Reconstruction by Serially Acquired 2D Slices Using a Distance Transform-Based Global Cost Function, Lecture Notes In Computer Science, Vol. 2308, pp. 390–400, 2002.

[11] M. Fano, M. Pol?k, The 3D Object Reconstruction from 2D Slices, Central European Seminar on Computer Graphics, CESCG 2000.

[12] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, A Hybrid 3D Watershed Algorithm Incorporating Gradient Cues and Object Models for Automatic Segmentation of Nuclei in Confocal Image Stacks, Cytometry pp. 23–36, Wiley 2003.

[13] A. Brandt, J. Mann, M. Brodski, and M. Galun, A fast and accurate multilevel inversion of the radon transform, SIAM J. Appl. Math., 60(2), pp. 437–462, 2000.

[14] K. Mueller, Fast and accurate three-dimensional reconstruction from cone-beam projection data using algebraic methods, Ph.D. dissertation, Ohio State Univ., Columbus, OH, 1998.

[15] G. Iain, B. Ari, B. Olivier, L. Frank, S. Sebastian, T. Scott, Evolution of computer technology for fast cone beam backprojection, Computational Imaging V, Proceedings of the SPIE, Vol. 6498, pp. 64980R, 2007.

[16] S. Xiao, Y. Bresler, and D. C. Munson, Jr., Fast Feldkamp algorithm for cone-beam computer tomography, Image Processing, 2003. ICIP 2003. Vol. 2, pp. 819–822.

[17] T. Rodet, P. Grangeat and L. Desbat, Multichannel algorithm for fast 3D reconstruction, Physics in Medicine and Biology, Vol. 47, pp. 2659-2671(13), 2002.

[18] S. Basu, Y. Bresler, $O(N^3 logN)$ Backprojection Algorithm for the 3-D Radon Transform, Medical Imaging, IEEE Trans., Vol. 21, pp. 76–88, 2002.